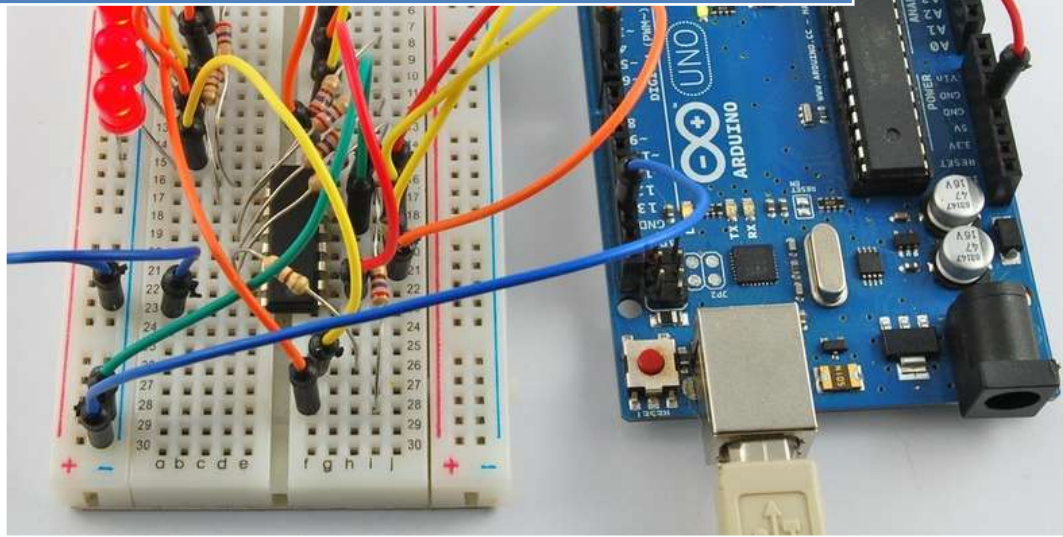


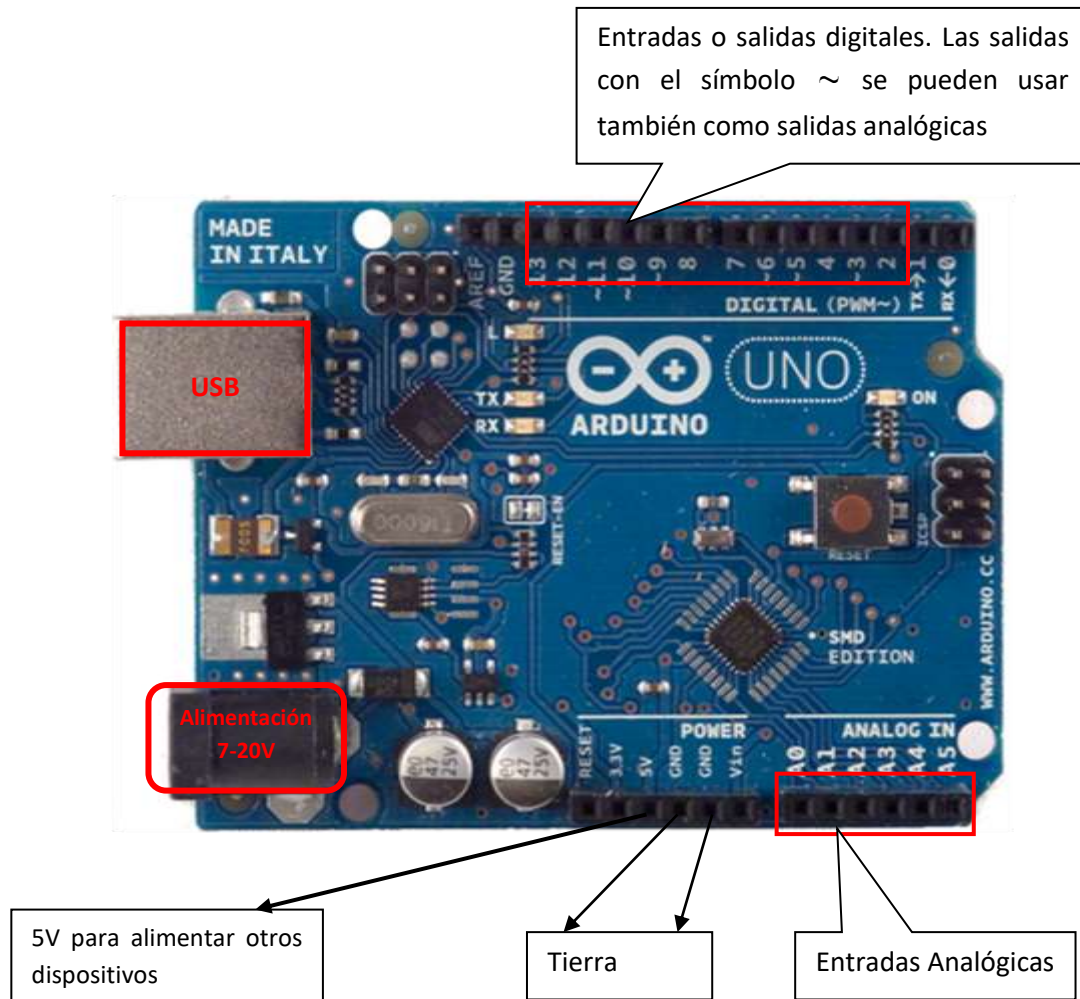
CONTROL AUTOMÁTICO CON ARDUINO



Señal analógica: es una señal que puede tomar infinitos valores

Señal digital: Es una señal que solo puede tomar dos valores 0 y 1 normalmente el 0 representa ausencia de tensión y el 1 presencia de tensión.

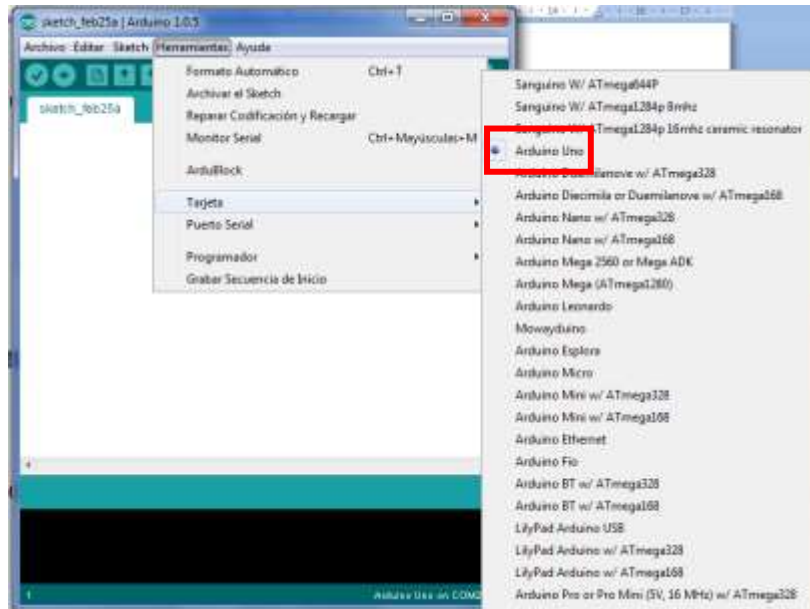
Arduino: consiste en una placa con un microcontrolador y una serie de entradas y salidas que pueden ser analógicas y digitales. Permite el desarrollo de múltiples diseños.



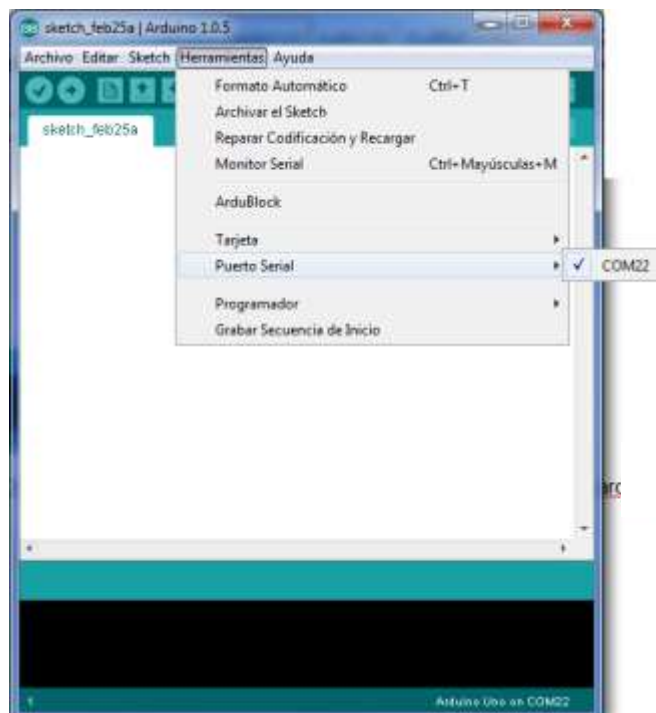
Para controlar las entradas y salidas del arduino utilizamos un programa

PRIMEROS PASOS PARA PROGRAMAR ARDUINO

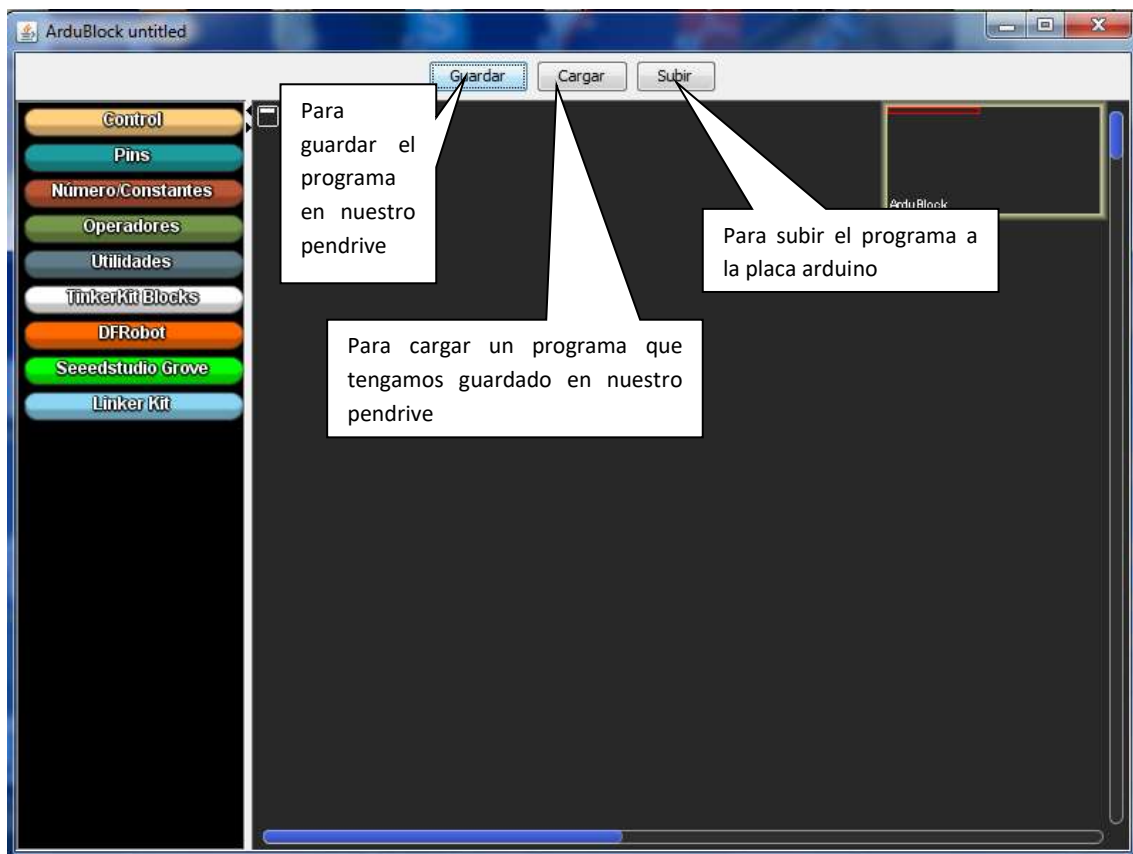
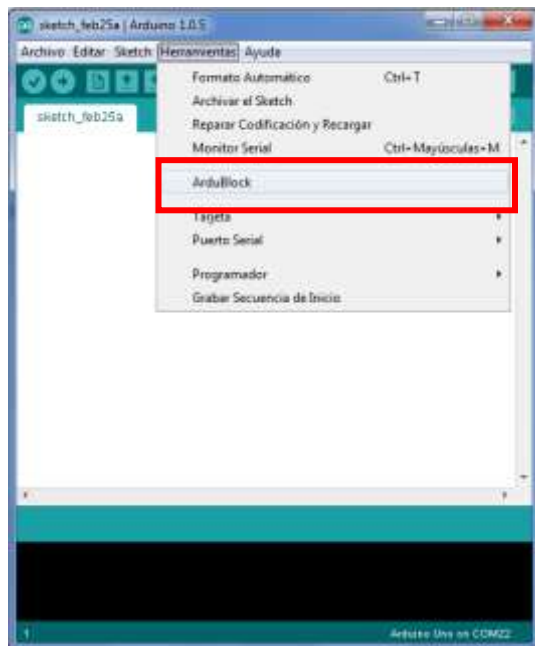
PRIMER PASO selecciono el tipo de arduino que estoy utilizando, en nuestro caso ARDUINO UNO.



SEGUNDO PASO: selecciono el puerto USB donde he conectado la placa arduino



TERCER PASO: entramos en ardublock



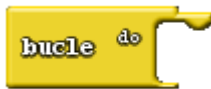
Los pasos uno y dos tenemos la opción de hacerlos después del tercero, ***pero siempre hay que hacerlos antes de presionar el botón de subir el programa a la placa arduino.***

REALIZACION PROGRAMAS PARA ARDUINO

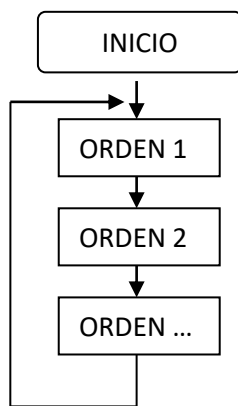
Para la realización de estos programas utilizaremos ardublock que nos permite realizar programas como si estuviésemos montando un puzle.

PIEZAS DE ARDUBLOCK

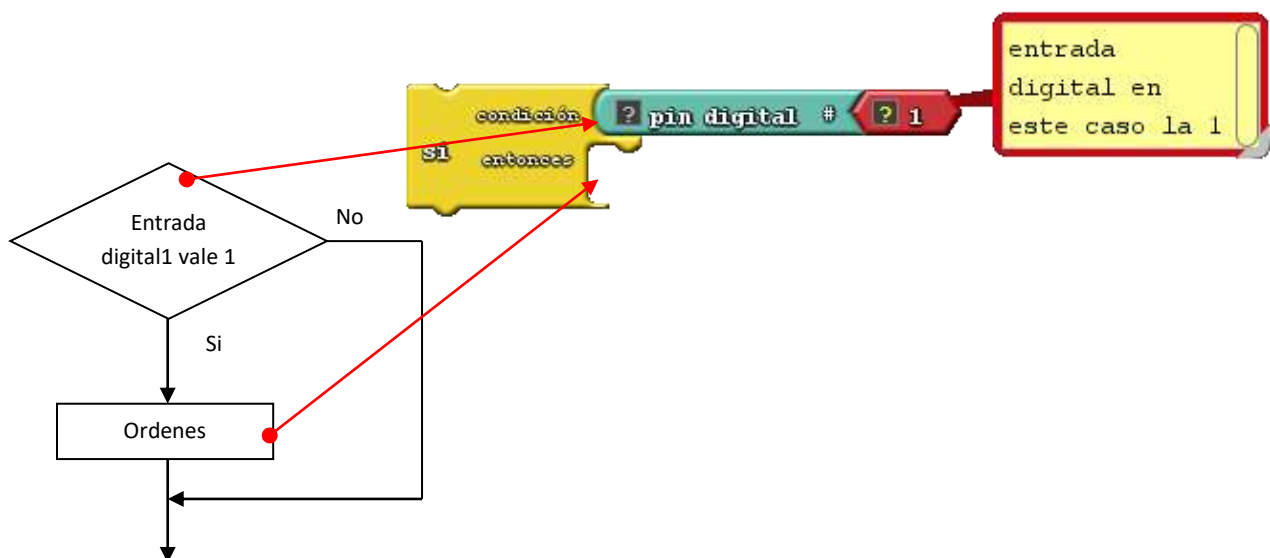
CONTROL

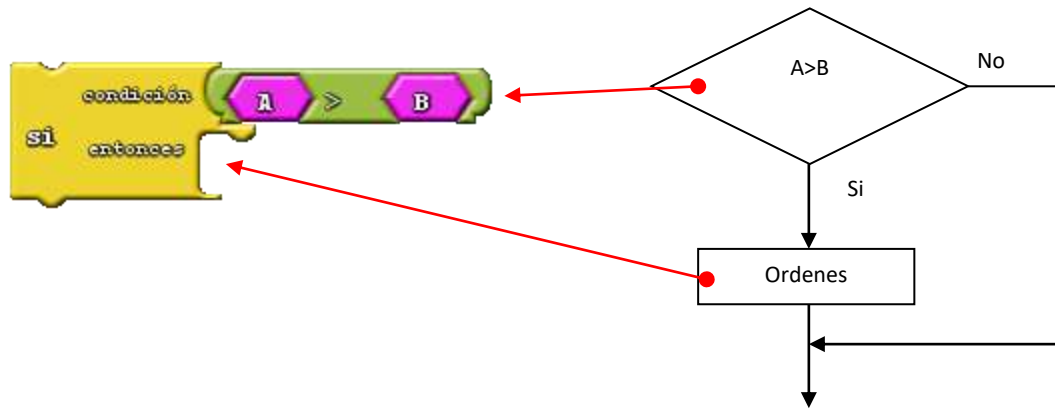


esta orden es con la que debemos empezar todos los programas, su funcionamiento es el siguiente: repite continuamente el paso por las ordenes que se encuentran dentro de él.

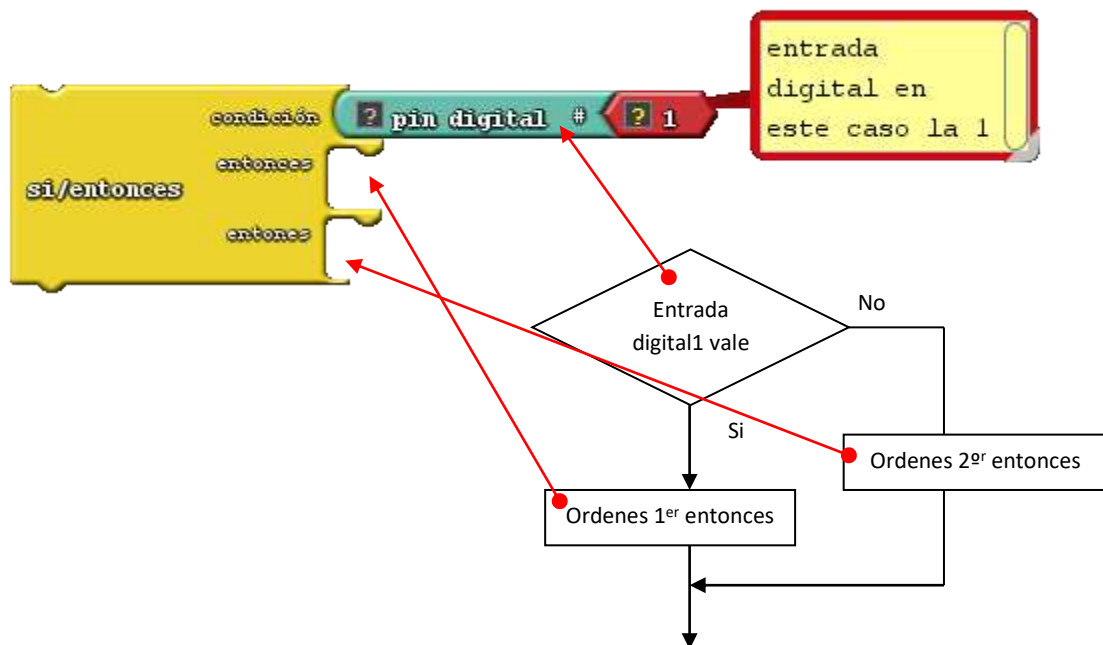


Si la **condición** es cierta (se cumple), se realizan las ordenes puestas dentro de **entonces**, si no se cumple las salta.





Si se cumple la condición hace las ordenes del 1^{er} entonces y salta las del segundo, Si no se cumple la condición hace las ordenes del 2^o entonces y se salta el 1^o

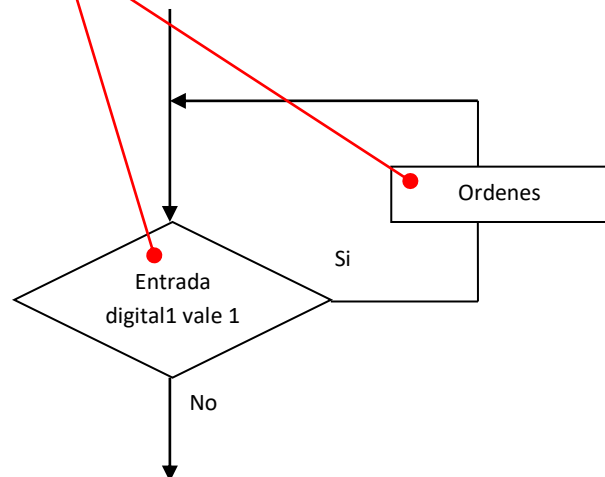




Repita las órdenes dentro de él mientras la condición sea cierta.



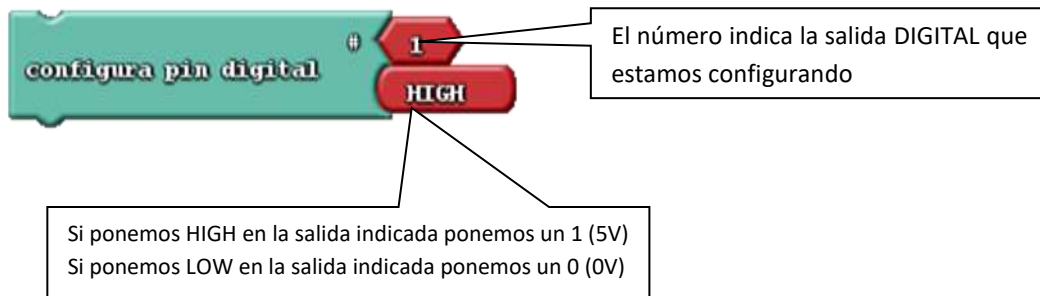
Mientras la entrada 1 en este caso valga 1 hará las órdenes que estén dentro, cuando la entrada 1 valga 0 seguirá



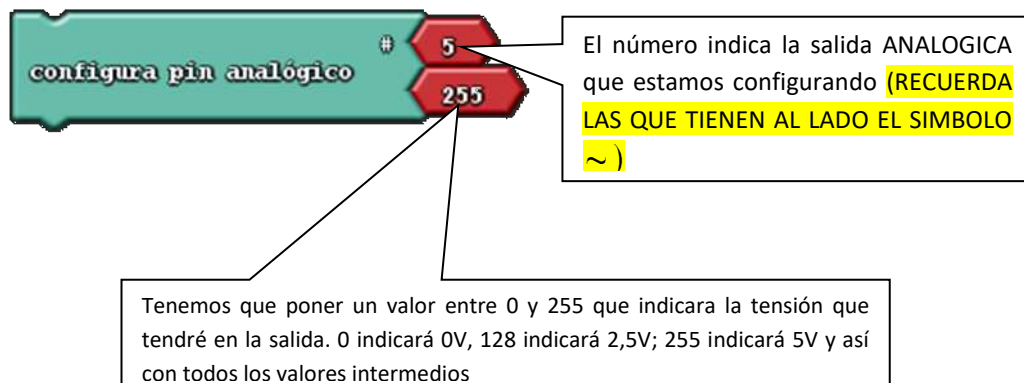
Repita las órdenes que tiene dentro el número de veces indicado

PINS

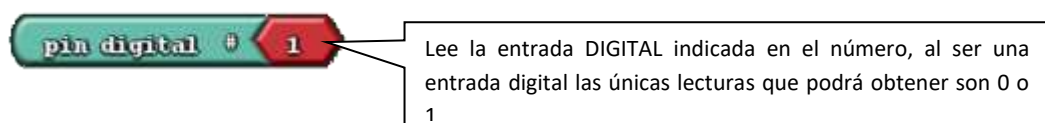
PONER VALOR A UNA SALIDA DIGITAL



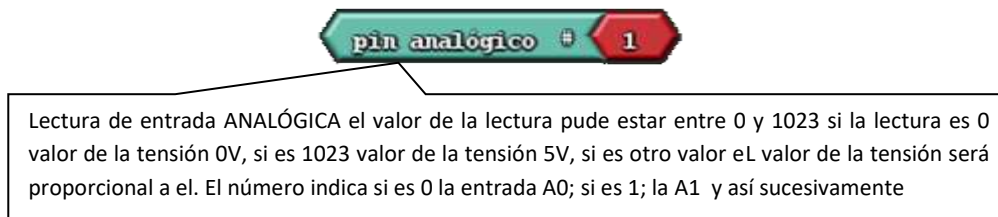
PONER VALOR A UNA SALIDA ANALOGICA



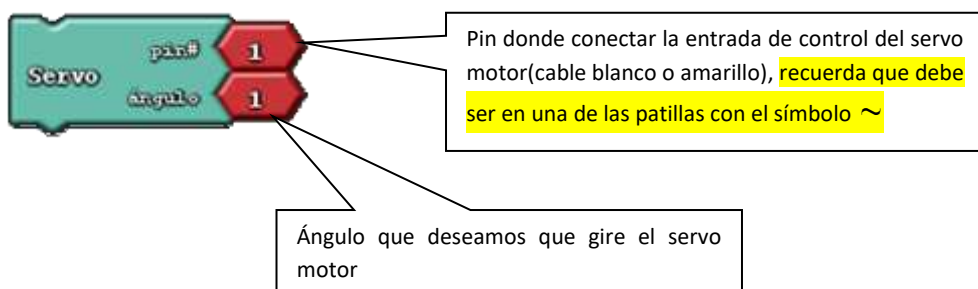
LECTURA ENTRADA DIGITAL



LECTURA VALOR ENTRADA ANALÓGICA

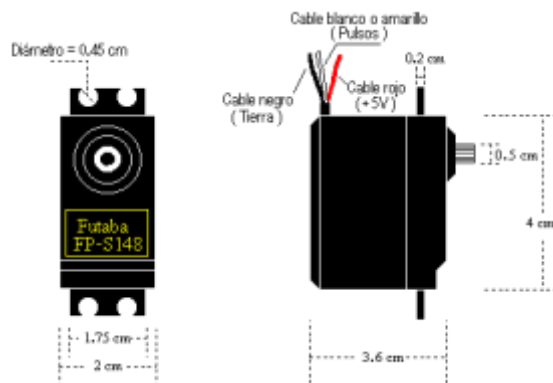


CONTROL SERVO MOTOR

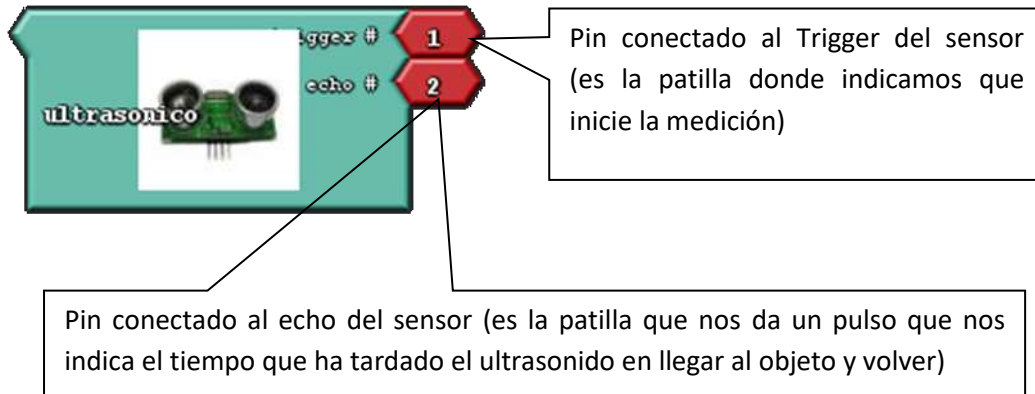


Un **servomotor** (también llamado **servo**) es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición.

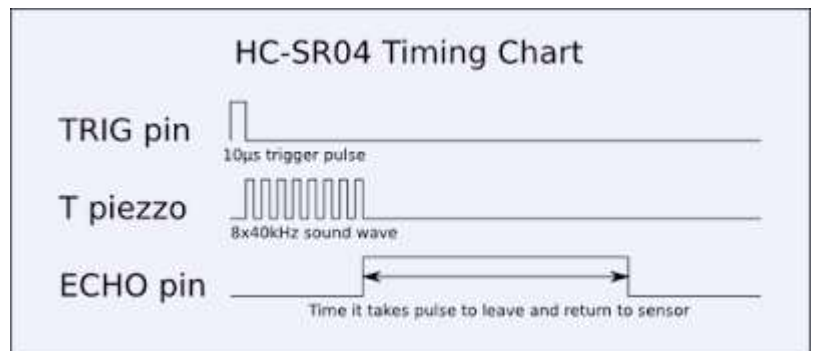
En otras palabras, un servomotor es un motor especial al que se ha añadido un sistema de control (tarjeta electrónica), un potenciómetro y un conjunto de engranajes. Con anterioridad los servomotores no permitían que el motor girara 360 grados, solo aproximadamente 180; sin embargo, hoy en día existen servomotores en los que puede ser controlada su posición y velocidad en los 360 grados. Los servomotores son comúnmente usados en modelismo como aviones, barcos, helicópteros y trenes para controlar de manera eficaz los sistemas motores y los de dirección.



OBTENCION DE DISTANCIA SENSOR DE ULTRASONIDOS



Nos da la distancia en cm



Funcionamiento:

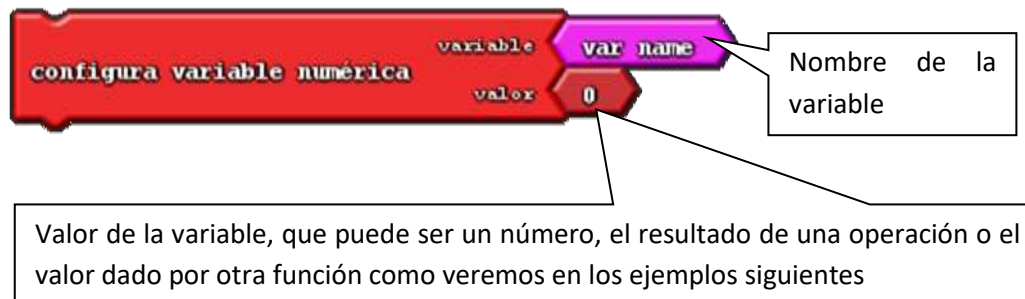
1. Enviar un Pulso "1" de al menos de 10µs por el Pin Trigger (Disparador).
2. El sensor enviará 8 Pulsos de 40KHz (Ultrasonido) y coloca su salida Echo a nivel alto (1), se debe detectar este evento e iniciar un conteo de tiempo.
3. La salida Echo se mantendrá en alto hasta recibir el eco reflejado por el obstáculo a lo cual el sensor pondrá su pin Echo a bajo, es decir, terminar de contar el tiempo.
4. Se recomienda dar un tiempo de aproximadamente 50ms de espera después de terminar la cuenta.

La distancia es proporcional a la duración del pulso y puedes calcularla con la siguiente formula (Utilizando la velocidad del sonido = 340m/s):

$$\text{Distancia en cm (centímetros)} = \text{Tiempo medido en } \mu\text{s} \times 0.017$$

NÚMEROS Y CONSTANTES

Una variable es un espacio de memoria, que definimos con un nombre, donde guardar información. En ardublock tenemos dos tipos de variables, las variables numéricas que como su nombre indica se usan para guardar números y las variables digitales donde solo podremos guardar dos valores (0 o LOW o FALSO) y (1 o HIGH o CIERTO).

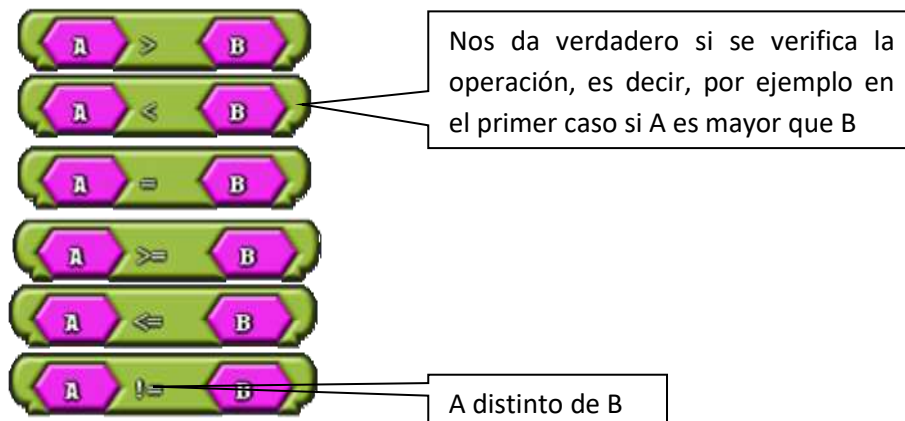
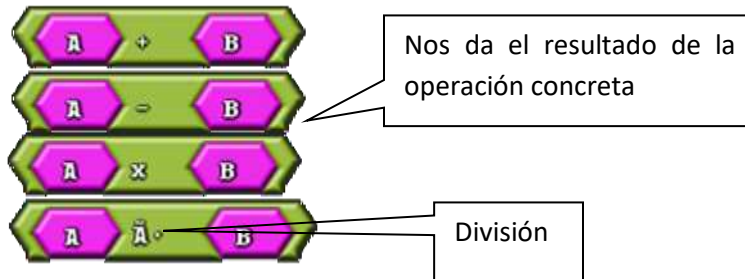


Aquí en la variable Distancia estamos guardando la distancia obtenida por el sensor de ultrasonidos



Aquí la variable contador la aumentamos en una unidad. Típico ejemplo de cómo hacer un contador

OPERADORES



UTILIDADES



Espera para realizar la siguiente orden tantos milisegundos como indique el número



Idéntico a la anterior pero ahora microsegundos



Nos devuelve un número obtenido aleatoriamente entre 0 el número colocado en la instrucción en este caso 1024



Nos permite convertir un rango de variación en otro. Realiza la siguiente conversión.

$$Valor\ map = \frac{valor \cdot valor\ sup.\ rango\ a}{valor\ sup.\ rango\ de}$$

$$Valor\ map = \frac{valor \cdot 255}{1023}$$

Ejemplo:

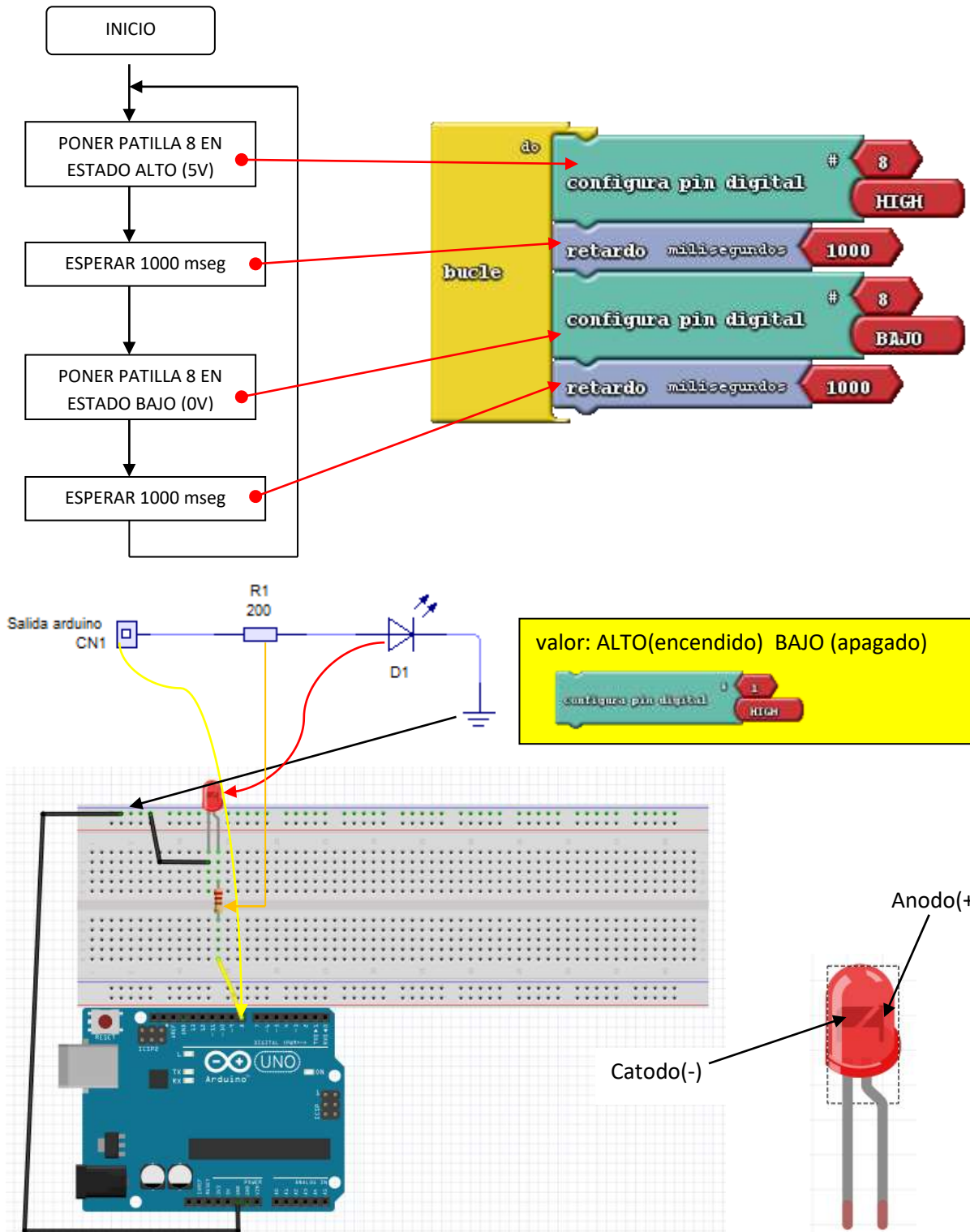
Si en valor tengo 0 me devolverá 0

Si en valor tengo 1023 me devolverá 255

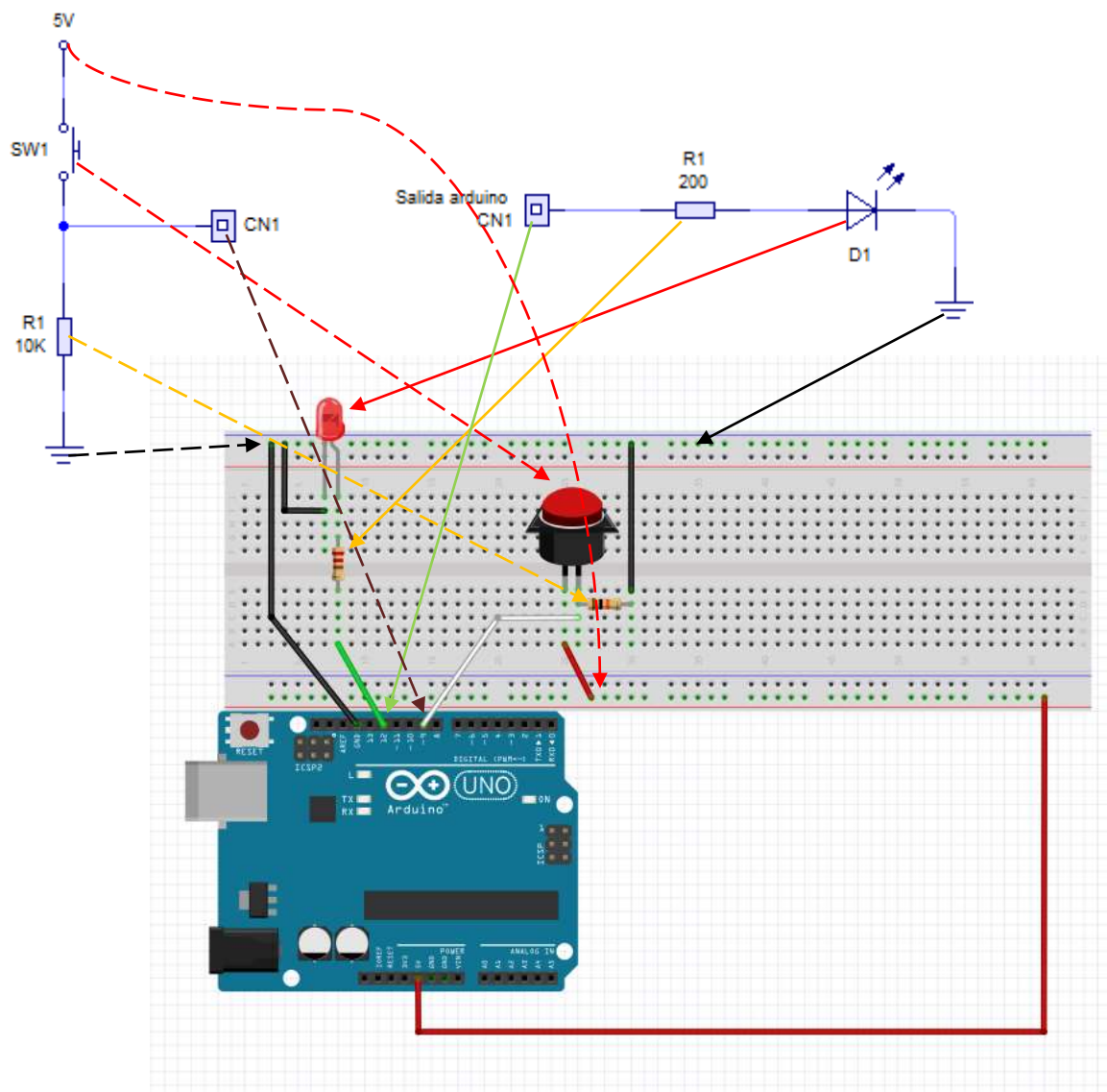
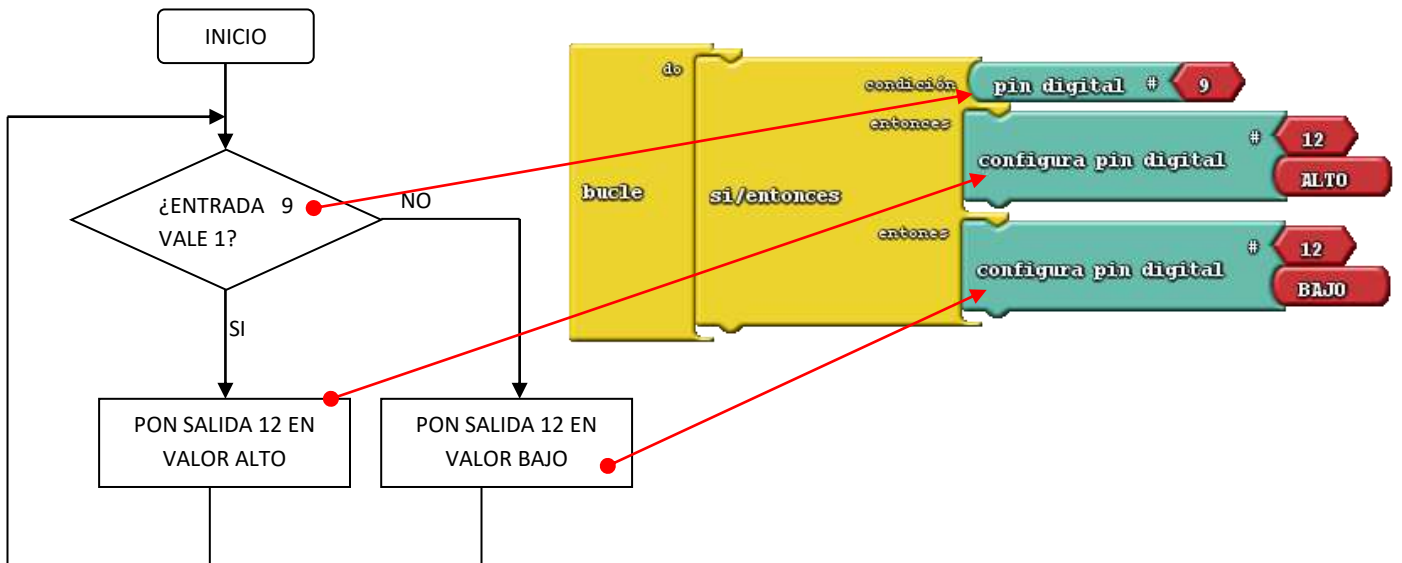
Si en valor tengo 511 me devolverá 127

Ejemplo de un programa

Hacer que un LED parpadee con un intervalo de un segundo, es decir que se encienda y se apague cada segundo. Conectaremos el LED a la patilla 8 del arduino



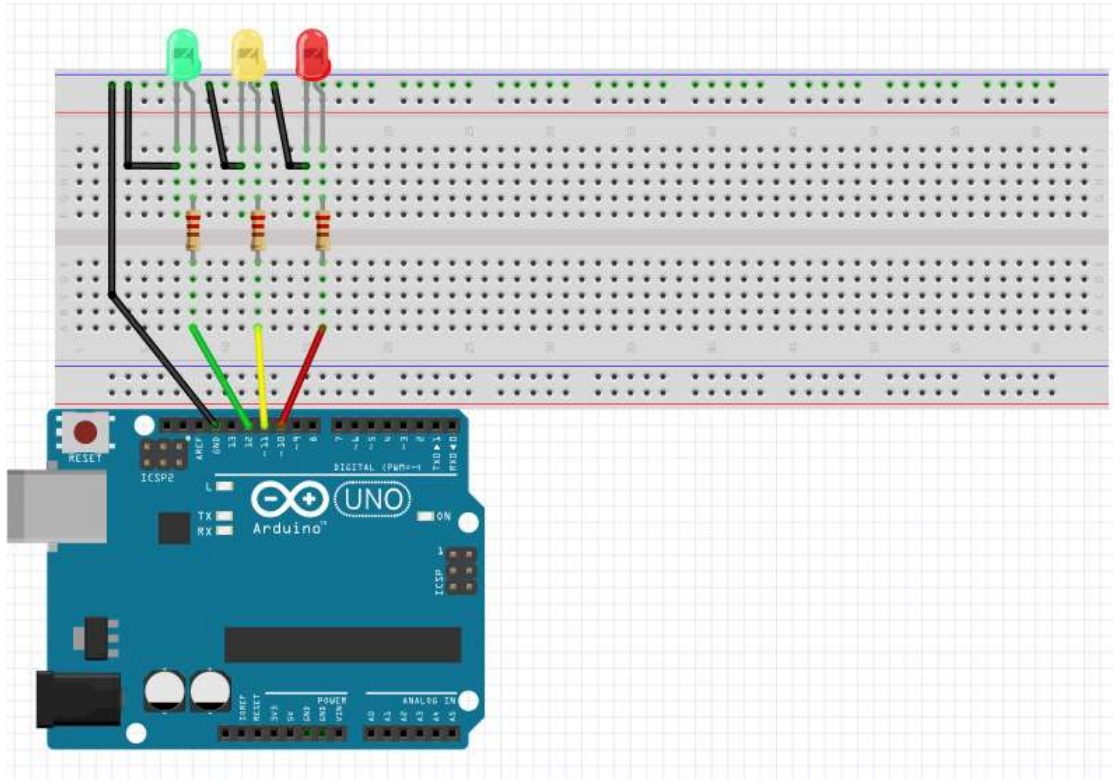
Realiza un programa que al presionar el pulsador se encienda un LED y al dejar de pulsar se apague.



EJERCICIOS

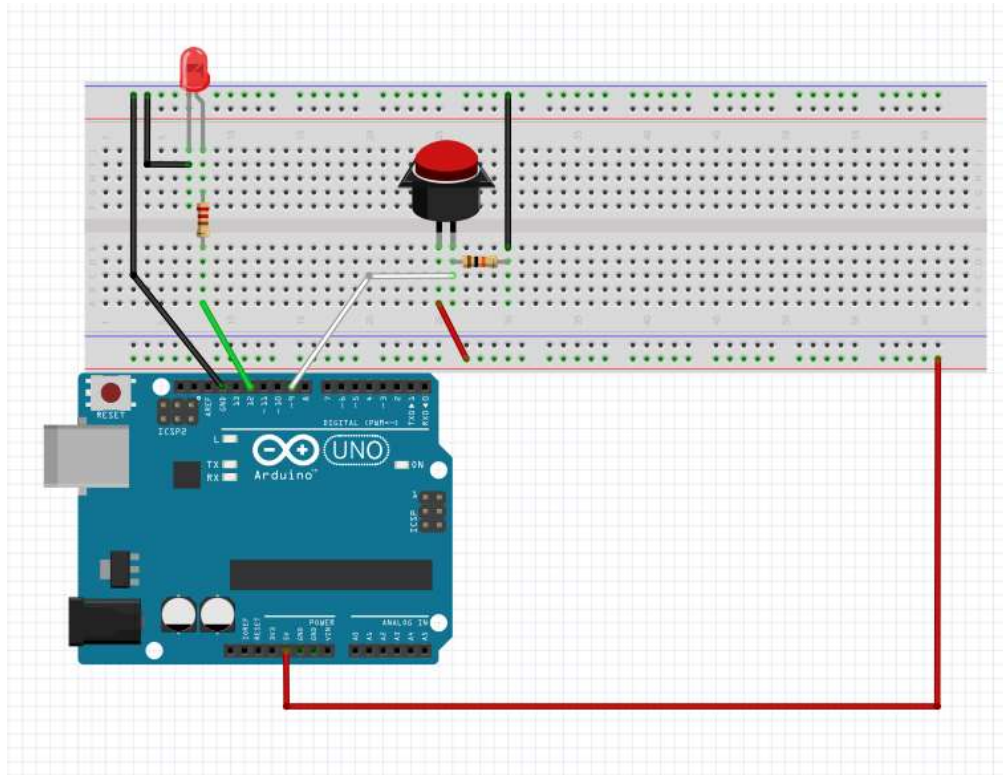
1.- Realiza un programa que controle 3 LEDs actuando como un semáforo. Luz verde encendida durante 5 segundos, amarilla 1.6 segundos y luz roja 4 segundos.

2.- Realiza un programa que controle 3 LEDs actuando como un semáforo. Luz verde durante 5 segundos, amarilla 2 segundos y luz roja 4 segundos. La luz amarilla debe parpadear con un periodo de 0,2 segundos.

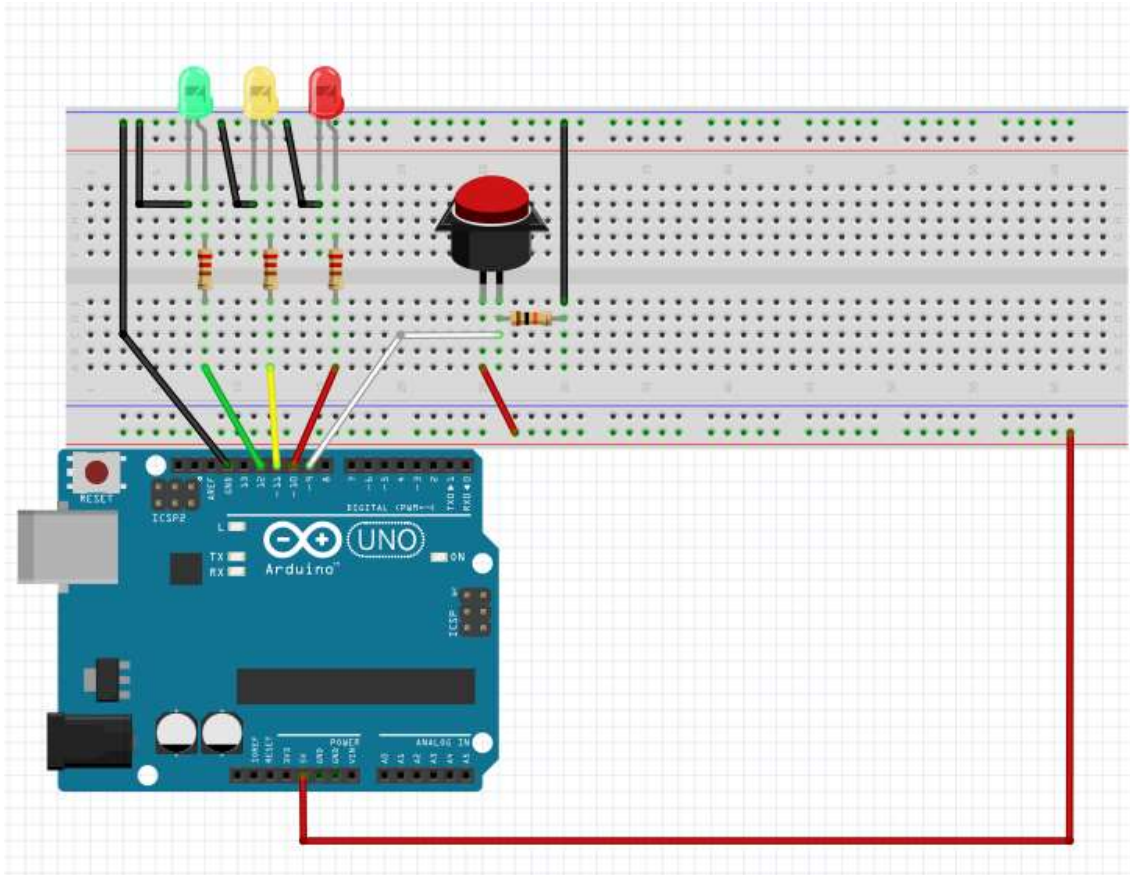


3.- Realiza un programa que al presionar el pulsador se encienda y apague un LED con un intervalo de 1 segundo y al dejar de pulsarlo se apague el LED. Realiza una variante, al pulsar la primera vez se encienda y apague el LED según la secuencia anterior al volverlo a pulsar que se apague

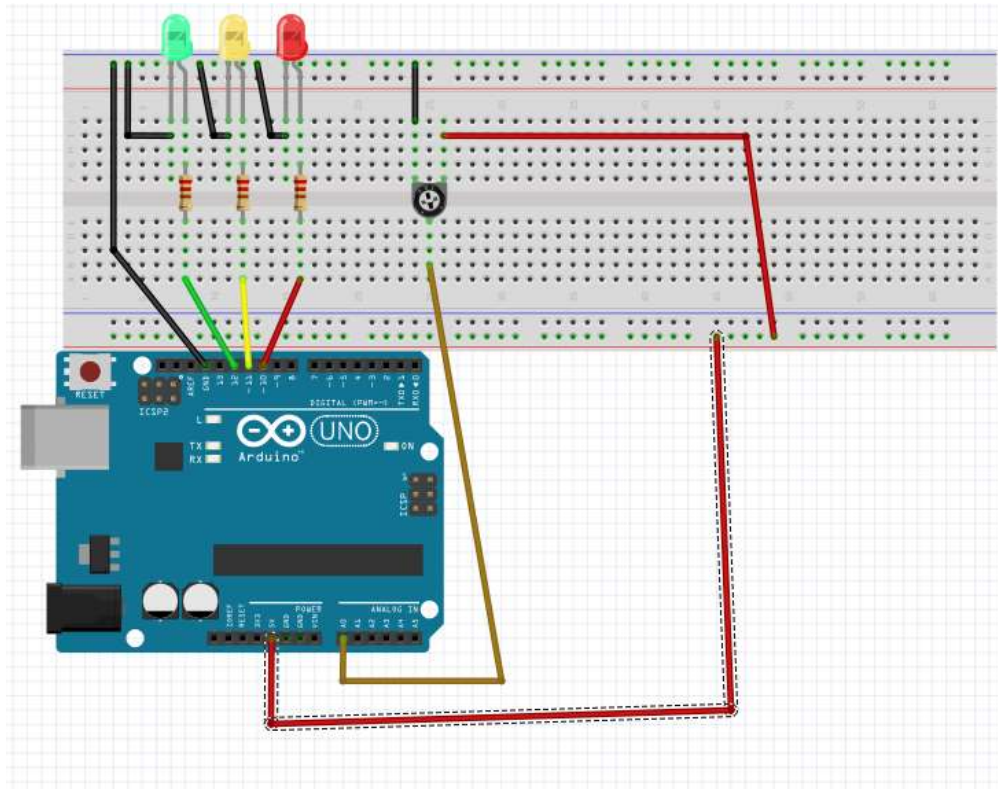
4.- Realiza un programa que al presionar el pulsador 10 veces se encienda y apague un LED 5 veces con un intervalo de 2 segundos.



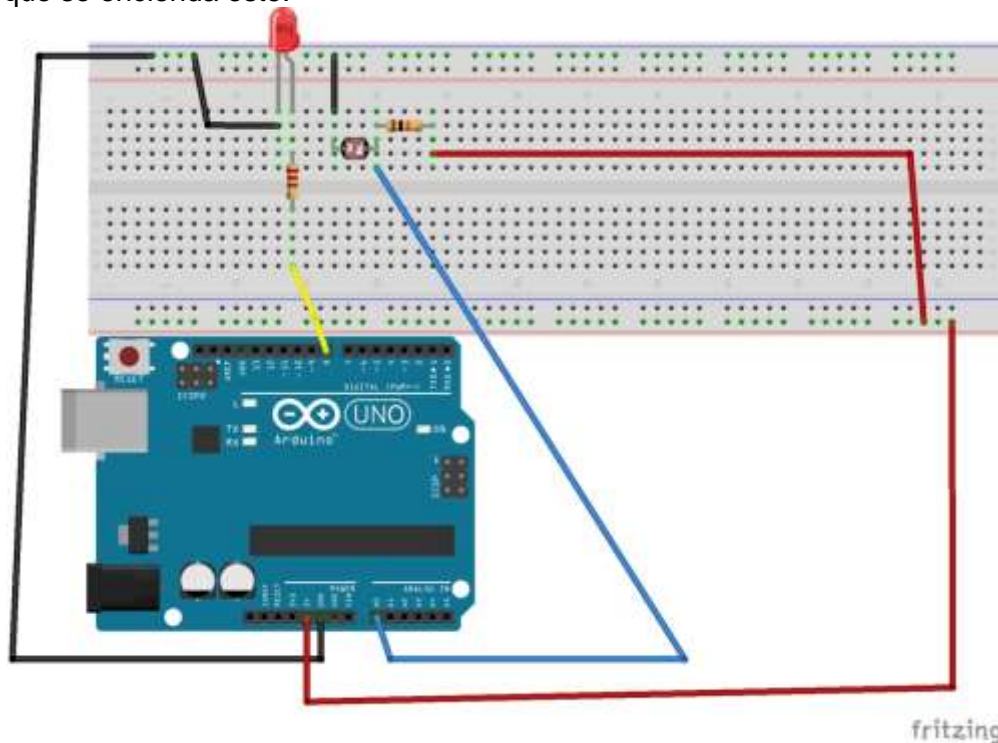
5.- Realiza un semáforo en el que los peatones tengan un pulsador que hace que al pulsarlo a los 5 segundos , el semáforo vaya a amarillo 2 segundos, a rojo después durante 5 segundos, al pasar este tiempo volverá a verde y se mantendrá así hasta que se vuelva a presionar el pulsador.



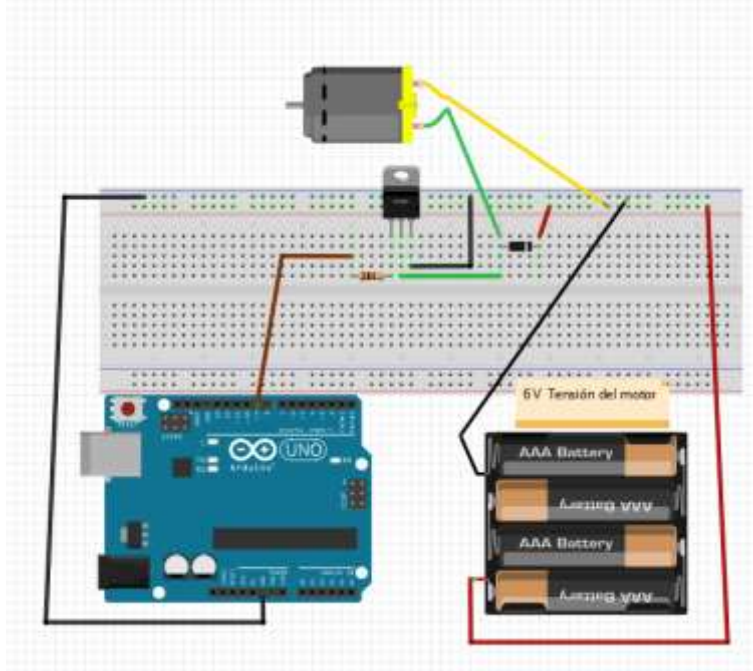
6.- Realiza un programa que al variar la resistencia de un potenciómetro se vayan activando tres LEDs. Simulando un vúmetro de tres LEDs.



7.- Programa que al incidir la luz sobre una LDR se apague un LED Y al oscurecerse que se encienda este.

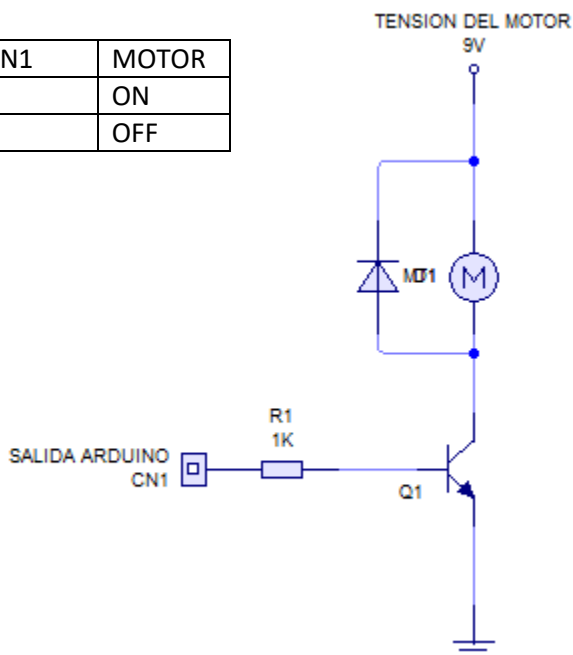


8.- Realiza un programa que active un motor de cc durante 5 segundos, lo pare 10 segundos y así sucesivamente (utilizando para ello un transistor)



Control de un motor cc mediante transistor

CN1	MOTOR
1	ON
0	OFF



Sin control de velocidad CN1 conectada a cualquiera de las salidas digitales del arduino
HIGH(1) O LOW (0)

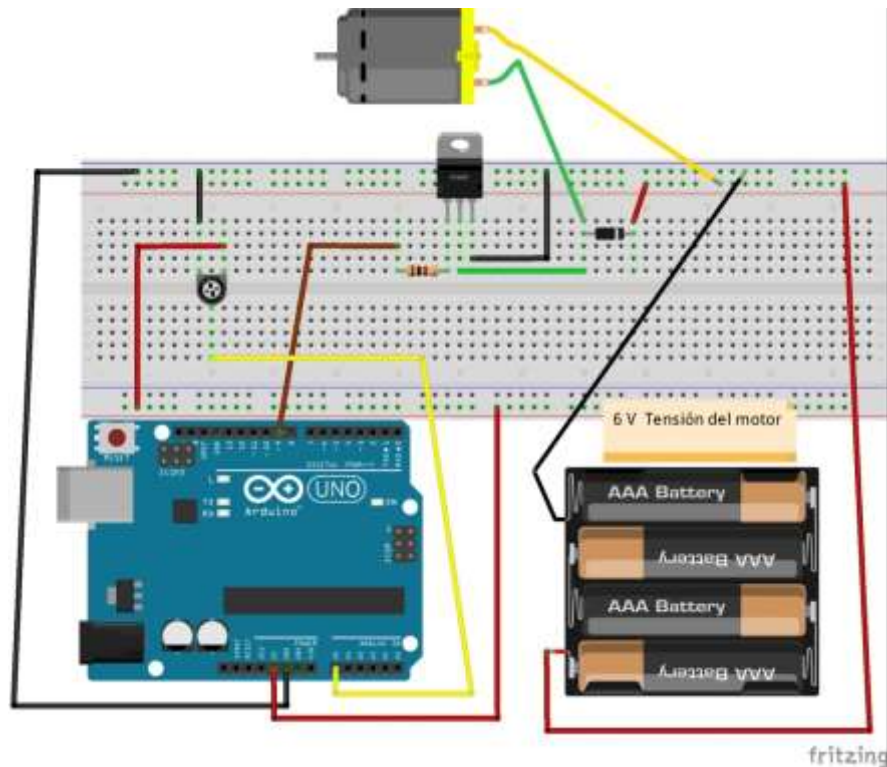


Con control de velocidad CN1 conectada a una salida PWM del arduino
value: valor entre 0 y 255

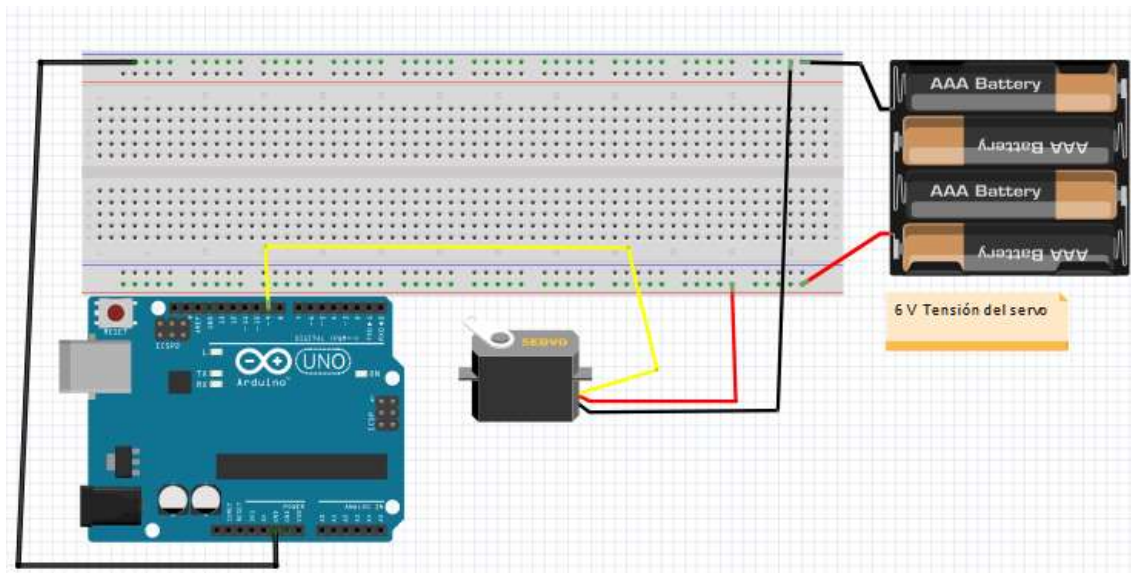


*PWM: Las siglas PWM vienen de "Pulse Width Modulation" (Modulación de Ancho de Pulso). Lo que hace este tipo de señal es emitir, en lugar de una señal continua, una señal cuadrada formada por pulsos de frecuencia constante (aproximadamente de 490 Hz). La gracia está en que al variar la duración de estos pulsos, estaremos variando proporcionalmente la tensión promedio resultante.

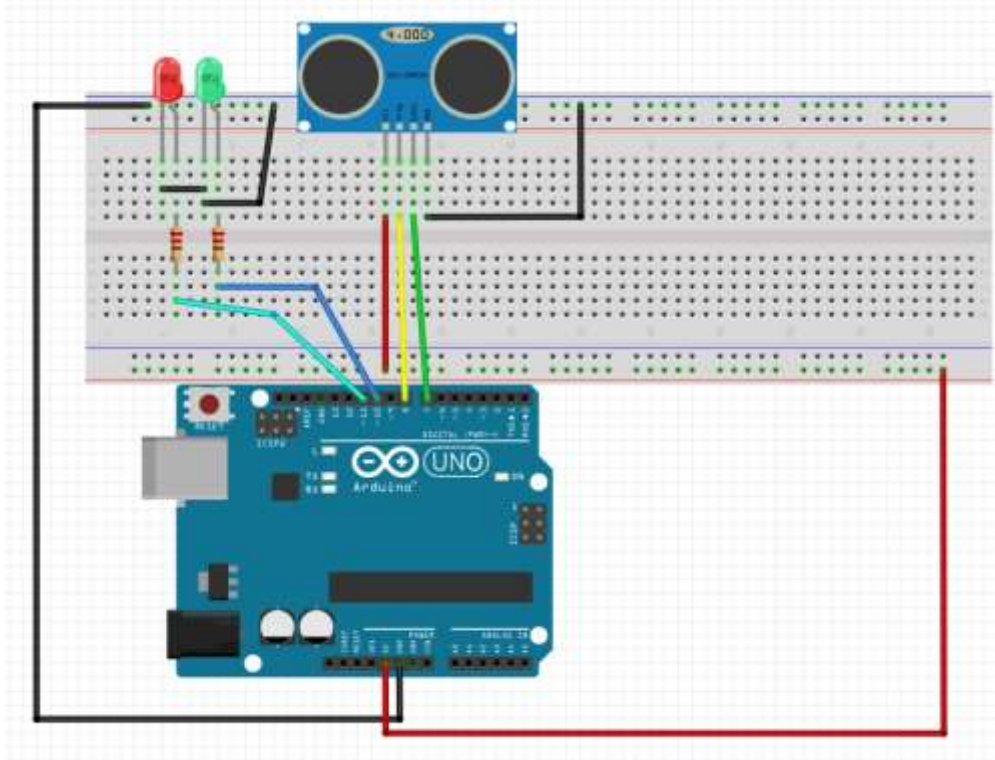
9.- Realiza un programa que al variar el valor de un potenciómetro, varíe la velocidad del motor (utiliza para ello un transistor)



10.- Realiza un programa que ponga un servo en posición extrema, 180°, espere 5 seg lo ponga en posición central 90° espere 5 seg lo lleve a posición 0° espere 5 seg lo lleve a 180° y repita la secuencia. Variante, Un programa que varíe el ángulo del servo dependiendo de la posición de un potenciómetro



11.- Realiza un sistema de aparcamiento de un coche de tal forma que si detecta un objeto a menos de 10 cm encienda un LED verde y uno rojo, si está entre 10,1 y 20 cm un LED rojo, si está entre 20,1 cm y 30 cm encienda un LED verde y si no se cumple ninguna de las condiciones los LEDs estarán apagados.



12.- El sistema cuenta con dos sensores (pulsadores) a la entrada y a la salida del parking, dos servomotores que controlan las barreras de entrada y salida, un LED rojo (parking completo) y un LED verde (parking libre). El parking tiene 10 plazas y su funcionamiento debe ser el siguiente: si hay plazas libres el LED verde debe estar activado y al presionar el pulsador de entrada se debe abrir la barrera de entrada durante 5 segundos y después bajarse, al presionar el pulsador de salida se debe abrir la barrera de salida durante 5 segundos y después bajarse. Si el parking está lleno se debe activar el LED rojo y aunque se presione el pulsador de entrada la barrera de entrada no se debe abrir.